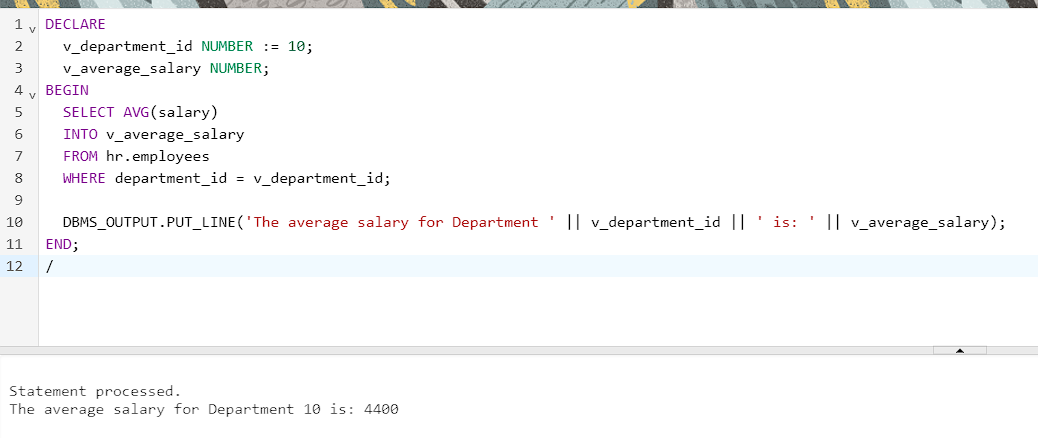
**Practice 3.1 (Interacting with the Oracle Server)**

1. Create an anonymous block to output the average salary for a particular department.



DECLARE

v\_department\_id NUMBER := 10;

v\_average\_salary NUMBER;

BEGIN

SELECT AVG(salary)

INTO v\_average\_salary

FROM hr.employees

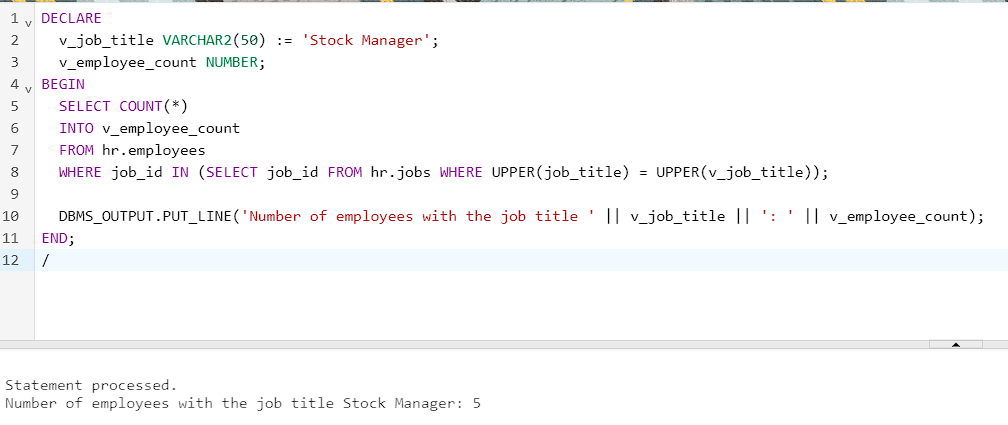
WHERE department\_id = v\_department\_id;

DBMS\_OUTPUT.PUT\_LINE('The average salary for Department ' || v\_department\_id || ' is: ' || v\_average\_salary);

END;

/

1. Create an anonymous block to count and output the number of employees of a particular position.



DECLARE

v\_job\_title VARCHAR2(50) := 'Stock Manager';

v\_employee\_count NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_employee\_count

FROM hr.employees

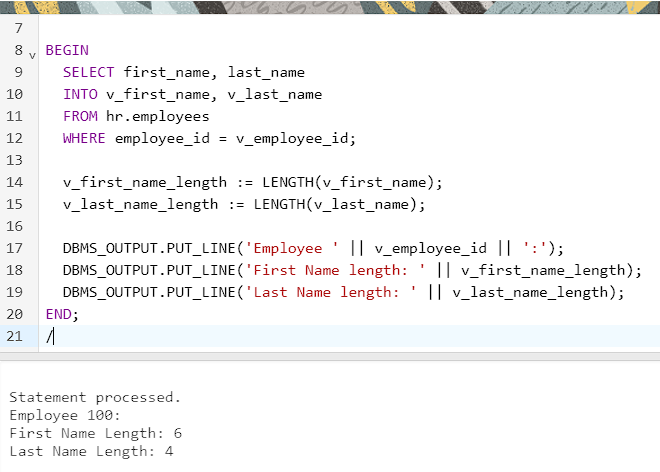
WHERE job\_id IN (SELECT job\_id FROM hr.jobs WHERE UPPER(job\_title) = UPPER(v\_job\_title));

DBMS\_OUTPUT.PUT\_LINE('Number of employees with the job title ' || v\_job\_title || ': ' || v\_employee\_count);

END;

/

1. Create an anonymous block to count and output the number of letters in the last name and the first name of a particular employee.



DECLARE

v\_employee\_id NUMBER := 100;

v\_first\_name VARCHAR2(50);

v\_last\_name VARCHAR2(50);

v\_first\_name\_length NUMBER;

v\_last\_name\_length NUMBER;

BEGIN

SELECT first\_name, last\_name

INTO v\_first\_name, v\_last\_name

FROM hr.employees

WHERE employee\_id = v\_employee\_id;

v\_first\_name\_length := LENGTH(v\_first\_name);

v\_last\_name\_length := LENGTH(v\_last\_name);

DBMS\_OUTPUT.PUT\_LINE('Employee ' || v\_employee\_id || ':');

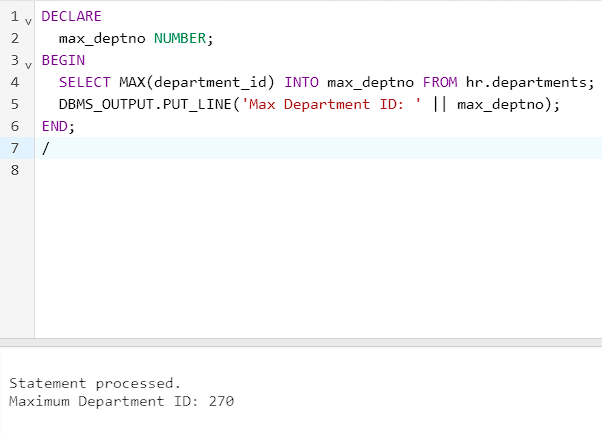
DBMS\_OUTPUT.PUT\_LINE('First Name length: ' || v\_first\_name\_length);

DBMS\_OUTPUT.PUT\_LINE('Last Name length: ' || v\_last\_name\_length);

END;

/

1. Create a PL/SQL block that selects the maximum department ID in the departments table and stores it in the max\_deptno variable. Display the maximum department ID.



DECLARE

max\_deptno NUMBER;

BEGIN

SELECT MAX(department\_id) INTO max\_deptno FROM hr.departments;

DBMS\_OUTPUT.PUT\_LINE('Max Department ID: ' || max\_deptno);

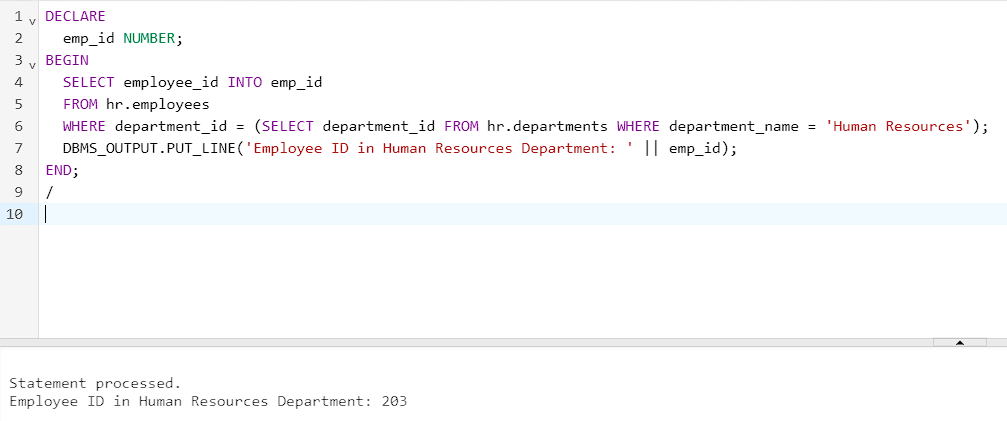
END;

/

1. Modify the PL/SQL block you created in exercise 4 and insert a new department with name ‘Education’ into the departments table. Use the SQL attribute SQL%ROWCOUNT to display the number of rows that are affected. Execute a select statement to check if the new department is inserted.

6. In exercise 5, you have set location\_id to null. Create a PL/SQL block that updates the location\_id to 3000 for the new department. Use a SELECT statement to display the department that you updated. Finally, include a DELETE statement to delete the department that you added.

7. Create a block with a single SELECT statement, which retrieves the employee\_id of the employee working in the ‘Human Resources’ department.



DECLARE

emp\_id NUMBER;

BEGIN

SELECT employee\_id INTO emp\_id

FROM hr.employees

WHERE department\_id = (SELECT department\_id FROM hr.departments WHERE department\_name = 'Human Resources');

DBMS\_OUTPUT.PUT\_LINE('Employee ID in Human Resources Department: ' || emp\_id);

END;

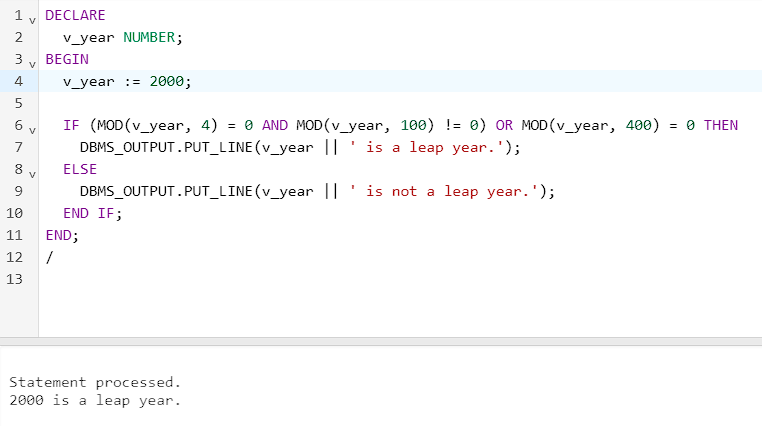
/

**Practice 3.2 (Writing Executable Statements)**

1. Write a PL/SQL block to accept a year and check whether it is a leap year. For example, if the year entered is 1990, the output should be “1990 is not a leap year.” Hint: The year should be exactly divisible by 4 but not divisible by 100, or it should be divisible by 400.

Test your solution with the following years:





DECLARE

v\_year NUMBER;

BEGIN

v\_year := 2000;

IF (MOD(v\_year, 4) = 0 AND MOD(v\_year, 100) != 0) OR MOD(v\_year, 400) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE(v\_year || ' is a leap year.');

ELSE

DBMS\_OUTPUT.PUT\_LINE(v\_year || ' is not a leap year.');

END IF;

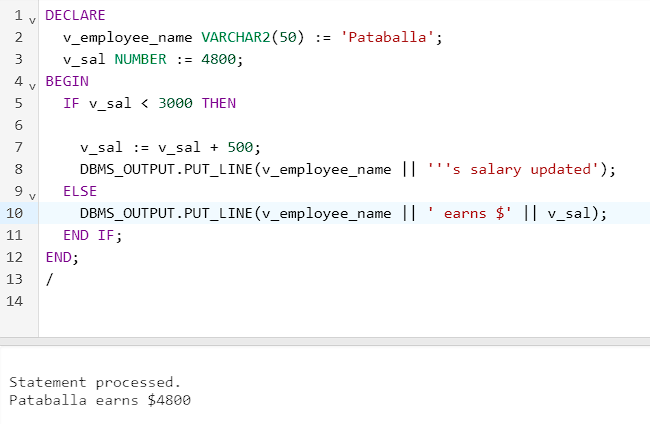
END;

/

2. Write a PL/SQL block to declare a variable called sal to store the salary of an employee. If the salary is less than 3,000, give the employee a raise of 500 and display the message “<Employee Name>’s salary updated” in the window. If the salary is more than 3,000, print the employee’s salary in the format, “<Employee Name> earns …...………”

Test the PL/SQL block for the following last names:





DECLARE

v\_employee\_name VARCHAR2(50) := 'Pataballa';

v\_sal NUMBER := 4800;

BEGIN

IF v\_sal < 3000 THEN

v\_sal := v\_sal + 500;

DBMS\_OUTPUT.PUT\_LINE(v\_employee\_name || '''s salary updated');

ELSE

DBMS\_OUTPUT.PUT\_LINE(v\_employee\_name || ' earns $' || v\_sal);

END IF;

END;

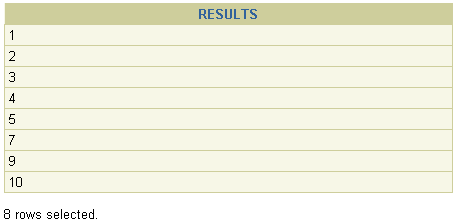
/

3. Create a table **messages** with one column Results of the type Number (3). Write a PL/SQL block to insert numbers into the messages table.

a. Insert the numbers 1 to 10, excluding 6 and 8.

b. Commit before the end of the block.

c. Execute a SELECT statement to verify that your PL/SQL block worked. You should see the following output. Use different types of loops.



CREATE TABLE messages (

Results NUMBER(3)

);

DECLARE

BEGIN

FOR i IN 1..10 LOOP

IF i <> 6 AND i <> 8 THEN

INSERT INTO messages (Results) VALUES (i);

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('FOR loop:');

FOR i IN (SELECT Results FROM messages) LOOP

DBMS\_OUTPUT.PUT\_LINE(i.Results);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('WHILE loop:');

DECLARE

v\_counter NUMBER := 1;

BEGIN

WHILE v\_counter <= 10 LOOP

IF v\_counter <> 6 AND v\_counter <> 8 THEN

SELECT Results INTO v\_counter FROM messages WHERE Results = v\_counter;

DBMS\_OUTPUT.PUT\_LINE(v\_counter);

END IF;

v\_counter := v\_counter + 1;

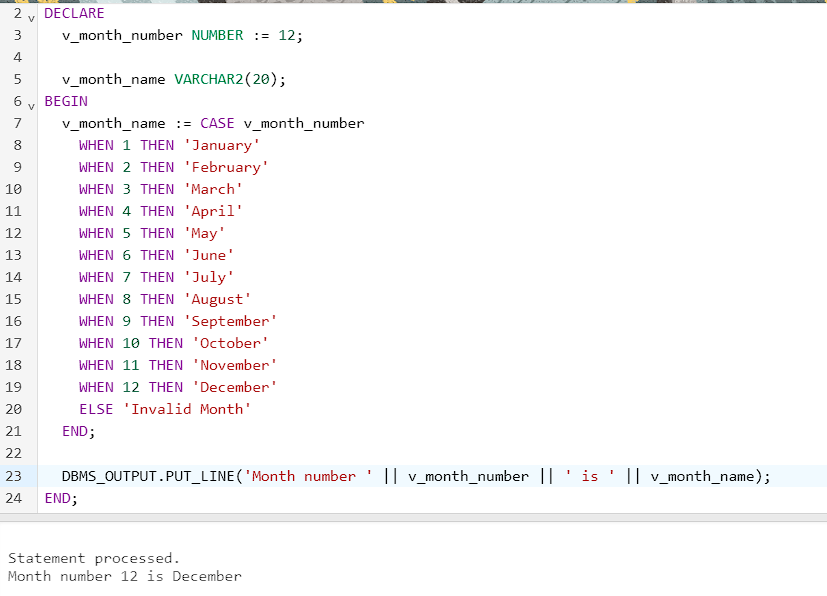
END LOOP;

END;

END;

/

4. Create a PL/SQL block to identify a name of the month depending on its number. For example, if a user input number 3, the output must be March. Use CASE expression and CASE statement control structures.



DECLARE

v\_month\_number NUMBER := 12;

v\_month\_name VARCHAR2(20);

BEGIN

v\_month\_name := CASE v\_month\_number

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

ELSE 'Invalid Month'

END;

DBMS\_OUTPUT.PUT\_LINE('Month number ' || v\_month\_number || ' is ' || v\_month\_name);

END;

/